The Korean Communications
in Statistics Vol. 9, No. 1, 2002
pp. 155-166

Bayesian Analysis for Neural Network Models

Younshik Chung[1], Jinhyouk Jung[2] and Chansoo Kim[3]

Abstract

Neural networks have been studied as a popular tool for classification and they are very flexible. Also, they are used for many applications of pattern classification and pattern recognition. This paper focuses on Bayesian approach to feed-forward neural networks with single hidden layer of units with logistic activation. In this model, we are interested in deciding the number of nodes of neural network model with p input units, one hidden layer with m hidden nodes and one output unit in Bayesian setup for fixed m. Here, we use the latent variable into the prior of the coefficient regression, and we introduce the 'sequential step' which is based on the idea of the data augmentation by Tanner and Wong(1987). The MCMC method(Gibbs sampler and Metropolish algorithm) can be used to overcome the complicated Bayesian computation. Finally, a proposed method is applied to a simulated data.

Keywords : Neural network, Latent variable, Sequential step, Gibbs sampler, Metropolish algorithm, Transfer function.

1. Introduction

Neural networks have been developed rapidly and now used in engineering applications widely. These models are typically presented as black box models to deal with nonlinear features in programs like regression, forecasting and classification.

Bayesian approach in the analysis of neural network models has been studied by Buntine and Weigend(1991), MacKay(1992), Neal(1996) and Muller and Rios Insua(1998a). From a statistical modeling point of view of neural networks are a special instance of mixture models. Many issues about posterior multimodality and computational stratiges in neural network modeling are of the relevance in the wider class of mixture model.

Linear regression which using more explanatory variables may give a better fit for the

1) Professor,Department of Statistics and Research Institute of Computer and Information Communication, Pusan National University, Pusan, 609-735 Korea
E-mail : yschung@hyowon.pusan.ac.kr
2) DB Marketing Team, Sejung Co., Bugogdong, Kumjunggu, Pusan, 609-735 Korea
3) Research Institute of Computer and Information Communication, Pusan National University, Pusan, 609-735 Korea

data, but may lead to overfitting and bad predictive performance. Similarly, increasing the size of a neural network may lead to better fits on training data. But, it may result in overfitting and poor predictions. Thus one needs a method for deciding how to choose a best model, and needs a way of searching the model space to find this best model, as it may be impossible to try fitting all possible models.

This paper is meant to address these issues and focuses on feed-forward neural networks with single hidden layer of units with logistic activation function. Our aim is to find the number of nodes of neural network model with p input units, one hidden layer with m hidden nodes and one output unit in Bayesian setup for fixed m.

Our methodology for this sequential step is based on the idea of the data augmentation by Tanner and Wong(1987). In our neural network model, we consider, by introducing the latent variable into the prior of the coefficient regression, the sequential step for the model selection which means that the model can take the β_j only after $\beta_1, \cdots, \beta_{j-1}$ are reached.

In other words, to get to the regressor, one must pass through the regressors $\beta_1, \cdots, \beta_{j-1}$.Then, the marginal posterior probabilities of the latent variables will be computed to decide the number of nodes. In orde to overcome the computational difficulties of marginal posterior density, we will use the Markov chain Monte Carlo (MCMC) methods such as Gibbs sampler(Gelfand and Smith, 1990)

and Metropolis-Hastings algorithm(Metropolis et al., 1953).

The plan of this article is as follows;

In section 2, we will briefly describe neural network model's origin, a kind of the transfer functions, and applications. Also, we will explain the model structure and interpretations of each variables. In section 3, we introduce how to construct the prior of latent variable w. To solve the computational difficulties in Bayesian approach, the MCMC method is employed. For this MCMC method, the full conditional densities are obtained. In section 4, we explain our methodology to a simulated data which has one hidden node. Finally in section 5, we discuss our results and propose directions for further works, and conclude this paper.

2. Neural Network Model

Neural networks were originally created as an attempt to model the act of thinking by modeling neurons in a brain. Much of the early work in this area traces back to a paper by McCulloch and Pitts(1943) which introduced the idea of an activation function, although the authors used a threshold(indicator) function rather than the sigmoidal functions common today. Threshold activations were found to have severe limitations and thus sigmoidal activations became widely used instead(Anderson 1982).

The unit combines its inputs into a single output value. This combination is called the

unit's *activation function*. There are three typical transfer function; the sigmoid, linear, and hyperbolic tangent functions. The specific values that the transfer function takes on are not as the general form of the function. The linear transfer function has limited practical value. A feed-forward neural network consisting only of units with linear transfer functions is really just doing a linear regression. The sigmoid and hyperbolic tangents are non-linear functions and result in non-linear behavior. The major difference between them is the range of their output. The most common transfer function is the S-shaped sigmoidal function. Even though it is not linear, the behavior of the sigmoid is appealing to statistician. In general, neural networks are a collection of simple computational units interlinked by a system of connections. Neural networks are used for many applications of pattern classification and pattern recognition. Many ideas and activities familiar to the statistician can be expressed in neural network notation. They include regression models from simple linear regression to projection pursuit regression, nonparametric regression(Specht, 1991), generalized additive models. Also many approaches are included to discriminant analysis such as logistic regression, classification trees. Neural network model's structure is composed of input units, hidden units and output units. We shall only consider feed-forward neural networks with one hidden layer of units with logistic activations and with one linear output unit. The Neural network diagram is as follows;

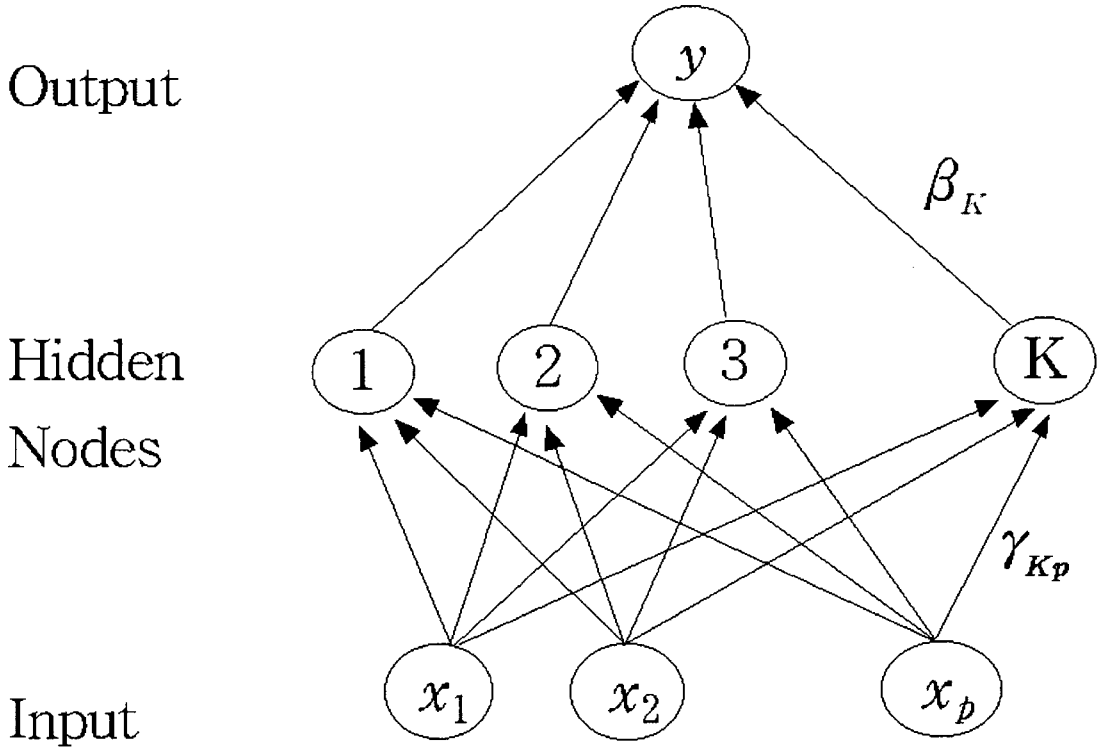

Figure 1. Neural network diagram

Feed-forward neural networks provide a flexible easy to a generalize linear regression functions. The model may be viewed as

$$y_i = \beta_0 + \sum_{j=1}^{m} \beta_j \Psi_j(\gamma_j' x_i) + \varepsilon_i, \quad i = 1, \cdots, n \tag{2.2}$$

and

$$\Psi(\gamma_j' x_i) = \frac{1}{1 + \exp(-\gamma_{j0} - \sum_{h=1}^{p} \gamma_{jh} x_{ih})} \tag{2.3}$$

where the error term has normal distribution with mean 0 and variance σ^2 , j is the index on the basis functions, known as hidden nodes, and the β_j's are the coefficients from the hidden nodes to the predicted responses. Ψ is logistic transformation of a linear combination of the explanatory variables γ_j which are the coefficients(weights) from the explanatory variables to the hidden nodes. The effect of the coefficients of the logistic basis functions (β) can be difficult to visualize because the logistic functions are nonlinear and can combine in unexpected ways. The coefficients inside the logistic function are even less interpretable. Now, we start by explaining the interpretations of the parameters for a model with only one hidden node; β_0 represents the overall location of y, as a sort of intercept, β_1 is the overall scale factor for y, the γ parameters control the location and scale the logistic function. Above model defines feed-forward neural networks with logistic activation functions, p input units, one hidden layer with m hidden nodes and one output node.

3. Bayesian Formulation

In this section, we introduce the Muller and Rios Insuas(1998b) model, who suggest a three-stage hierarchical model. The hierarchical structure is simple, although many parameters are multivariate. Muller and Rios Insua(1998b) consider the linear regression of a response y on covariates $x_1, \cdots, x_p$ by using a hidden layer of m nodes with logistic activation functions. This actually corresponds to the combination of two standard statistical models, linear and logistic regression. Densities are denoted generally by brackets. For example, [X,Y], [X|Y] and [X] are joint, conditional and marginal form, respectively. From (2.2), the distribution of output y_i can be expressed as

$$[y_i | \beta, \gamma, \sigma^2] \sim N(\beta_0 + \sum_{j=1}^{m} \beta_j \Psi(\gamma_j' x_i), \sigma^2), \quad i = 1, \cdots, n, \tag{3.1}$$

where $\beta = (\beta_0, \beta_1, , \cdots, \beta_m)'$, $\gamma_j = (\gamma_{j0}, \gamma_{j1}, \cdots, \gamma_{jp})'$, $j = 1, \cdots, m$, p is the number of

input units and m is the number of hidden nodes.

One approach for selecting a prior is to choose a prior that reflects one's beliefs. Such a prior would typically be a proper prior that integrates to one. Another possible approach would be to use a noninformative(flat) prior that does not favor particular values of the parameter over other values. For a small number of explanatory variables under consideration, one could try to find the best model using all possible subsets. That is, given the output y and input x, we find the "best" model which contain a selected subset $\beta_1, \cdots, \beta_k$, of $\beta_1, \cdots, \beta_m$ with $k \leq m$. However this becomes difficult as the number of variables increases. Further more, one would need to select an optimal network size for each subset. Thus an efficient algorithm is necessary for searching over the model space to find models of high posterior probability.

3.1. Hierarchical Mixture Model

The key of construction of Bayesian model selection in Neural network model is that each component of β is modeled as having come from a mixture of two normal distributions with different variances. By introducing the latent variable $w_j = 0$ or 1, we represent our normal mixture by

$$[\beta_j | w_j] \sim (1 - w_j) N(0, \tau_j^2) + w_j N(0, c_j^2 \tau_j^2), \quad j = 1, \cdots, m. \tag{3.2}$$

As will be seen, the introduction of w_j facilitates our analysis of the problem. Our methodology is based on the data augmentation idea of Tanner and Wong(1987). This setup is exactly same as George and McCulloch (1993) except the choice of prior of the latent variables in section 3.2. Recently, Chung and Kim (1999) used the similar setup to detect the outliers in regression model. When $w_j = 0$, $\beta_j \sim N(0, \tau_j^2)$ and when $w_j = 1$, $\beta_j \sim N(0, c_j^2 \tau_j^2)$. Our interpretation of this formulation is as follows. First, we set $\tau_i (> 0)$ small so that if $w_j = 0$ then β_j would probably be so small that it could be "safely" estimated by 0. Second, we set c_j large ($c_j > 0$ always) so that if $w_j = 1$, then a non-zero estimate of β_j should probably be included in the next model. To obtain (3.2) as the prior for $[\beta_j | w_j]$, we use a multivariate normal prior as follows;

$$[\beta | w] \sim N(0, D_w R D_w) \tag{3.3}$$

where $\beta = (\beta_1, \cdots, \beta_m)'$, $w = (w_1, \cdots, w_m)'$, R is the identity matrix, and

$$D_w \equiv diag[b_1 \tau_1, \cdots, b_m \tau_m] \tag{3.4}$$

with $b_j=1$ if $w_j=0$ and $b_j=c_j$ if $w_j=1$. D_w determines the scaling of the prior covariance matrix in such a way that (3.2) is satisfied. Here too, we set $\tau_1,\cdots,\tau_m$ small and $c_1,\cdots,c_m$ large ($c_j>1$ always) so that under (3.3), those β_j for which $\tau_j=0$ will tend to be clustered around 0, whereas those β_j for which $w_j=1$ will tend to be dispersed. The choice of c_j should be such that if $\beta_j\sim N(0,\ c_j\ \tau_j^2)$, then a non-0 estimate of β_j should be included in the final model. One would want to choose c_j large enough to give support to values of β_j that are substantively different from 0, but not so large that unrealistic values of β_j are supported. To help guide the choice of c_j, it may be useful to observe that the density of $N(0,\ c_j\ \tau_j^2)$ and $N(0,\ \tau_j^2)$ intersect at $\xi(c_j)\tau_j$ when $\xi(c_j)=\sqrt{2(\log c_j)c_j^2(c_j^2-1)}$. This implies that the density of $N(0,c_j^2\tau_j^2)$ will be larger than the density of $N(0,\tau_j^2)$ iff $|\beta_j|>\xi(c_j)\tau_j$. Note that this intersection point increases very slowly. For example, the choices c_j=10,100,1000,10000,100000 correspond to $\xi(c_j)\approx 2.1,\ 3.1,\ 3.7,\ 4.3,\ 4.8$. It may also be useful to observe that c_j is the ratio of the heights of $N(0,\tau_j^2)$ and $N(0,\ c_j^2\tau_j^2)$ at 0. Thus c_j can be interpreted as the prior odds that x_i should be excluded when β_j is very close to 0. Then the hierarchical structure is as follows;

$$[\gamma_j\,|\,\mu]\sim N_p(\mu,\ \sigma_\gamma^2 I),\ \ j=1,\cdots,m, \tag{3.5}$$

$$[\sigma^2]\sim IG(\frac{s}{2},\ \frac{S}{2}), \tag{3.6}$$

$$[\mu]\sim N_p(a_\gamma,\ A_\gamma), \tag{3.7}$$

and

$$w=(w_1,\cdots,w_m)\sim \pi(w) \tag{3.8}$$

where $\mu=(\mu_1,\cdots,\mu_p)'$, a_γ is $p\times 1$ column vector, and A_γ is $p\times p$ positive definite matrix.

3.2. Prior of Latent Variable

Our main reason for embedding the normal linear model (3.1) in the hierarchical mixture model is to obtain the marginal posterior distribution $f(w\,|\,y)\propto f(y\,|\,w)\pi(w)$, which contains the information relevant to the number of nodes. $\pi(w)$ may be interpreted as the statistician's prior probability that the $\Psi(\gamma_j' x_i)$'s corresponding to non-zero components of w.

In our neural network model, we consider the sequential step for our model selection

which means that the model can take the β_j only after $\beta_1, \cdots, \beta_{j-1}$ are reached. In other words, to get to the regressor β_j, one must pass through the regressors $\beta_1, \cdots, \beta_{j-1}$. Let $p_j = P_r(w_j | w_1, \cdots, w_{j-1})$. p_j can be determined when $w_1 = \cdots w_{j-1} = 1$, but if at least one of the values of $w_1, \cdots, w_{j-1}$ is zero, p_j must be zero. Therefore, the joint prior density $\pi(w)$ of w is in (3.8) is defined as follows;

Let $P(w_1 = 1) = p_1$ and $P(w_1 = 0) = 1 - p_1$. Since the value of β_2 is considered only when β_1 exists, set $P(w_2 = 1 | w_1 = 1) = p_2$ and $P(w_2 = 0 | w_1 = 1) = 1 - p_2$. Thus,

$P(w_2 = 0 | w_1 = 0) = 1$ and $P(w_2 = 1 | w_1 = 0) = 0$. Similarly, set $P(w_3 = 1 | w_2 = 1, w_1 = 1) = p_3$

$P(w_3 = 0 | w_2 = 1, w_1 = 1) = 1 - p_3$. Then, $P(w_3 = 0 | w_2 = 0, w_1 = 0) = 1$ and

$$P(w_3 = 1 | w_2 = 0, w_1 = 0) = P(w_3 = 1 | w_2 = 0, w_1 = 1) =$$
$$P(w_3 = 1 | w_2 = 1, w_1 = 0) = P(w_3 = 0 | w_2 = 1, w_1 = 0) = 0.$$

Therefore, the joint prior density $\pi(w)$ of w is of the general form

$$\pi(w_1, \cdots, w_m) = \left(\prod_{j=1}^{m} p_j \right)^{I(w_m = 1, \cdots, w_1 = 1)} \times \left((1 - p_m) \prod_{j=1}^{m-1} p_j \right)^{I(w_m = 0, w_{m-1} = 1, \cdots, w_1 = 1)}$$
$$\times \left((1 - p_{m-1}) \prod_{j=1}^{m-2} p_i \right)^{I(w_m = 0, w_{m-1} = 0, w_{m-2} = 1, \cdots, w_1 = 1)}$$
$$\times \cdots \times \left((1 - p_2) p_1 \right)^{I(w_m = 0, \cdots, w_2 = 0, w_1 = 1)} \times (1 - p_1)^{I(w_m = \cdots = w_1 = 0)} \quad (3.9)$$

where I(·) denote the indicator function.

3.3. Full conditional densities

Now, we will find the functional forms of marginal posterior distributions of the parameter. Then the joint posterior density of $\beta, \sigma^2, \gamma, \mu, w$ given y is given by

$$[\beta, \gamma, \sigma^2, \mu, w | y] = (2\pi\sigma^2)^{-\frac{n}{2}} \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{m} \beta_j \Psi(\gamma_j' x_i))^2 \right]$$
$$\times (2\pi)^{-\frac{m}{2}} |D_w R D_w|^{-\frac{1}{2}} \exp\left[-\frac{1}{2} \beta' (D_w R D_w)^{-1} \beta \right]$$
$$\times (2\pi)^{-\frac{m}{2}} \exp\left[-\frac{1}{2} (\gamma_j - \mu)' \sigma_\gamma^{-2} (\gamma_j - \mu) \right]$$
$$\times \frac{1}{(2\pi) |A_\gamma|^{\frac{1}{2}}} \exp\left[-\frac{1}{2} (\mu - a_\gamma)' A_\gamma^{-1} (\mu - a_\gamma) \right]$$
$$\times \frac{\left(\frac{S}{2}\right)^{\frac{s}{2}}}{\Gamma\left(\frac{s}{2}\right)} \left(\frac{1}{\sigma^2}\right)^{\frac{s}{2}+1} \exp\left[-\frac{S}{2\sigma^2} \right]$$

$$\times \pi(w) \tag{3.10}$$

where $w=(w_1,\cdots,w_m)'$ is defined in (3.9).

In order to apply Gibbs sampler, the full conditional distributions are needed as follows; for $j=1,\cdots,m$,

$$[\beta_j|\beta_{(j)}, w, \gamma, \mu, \sigma^2 y] = N(m_\beta, s_\beta^2) \tag{3.11}$$

where $\beta_{(j)}=(\beta_1,\beta_2,\cdots,\beta_{j-1},\beta_{j+1},\cdots,\beta_m)$, and

$$m_\beta = \frac{b_j^2\tau_j^2\sum_{i=1}^{n}\Psi(\gamma_j' x_i)(y_i-\beta_0-\sum_{j\neq k}^{m}\beta_j\Psi(\gamma_j' x_i))}{b_j^2\tau_j^2\sum_{i=1}^{n}\Psi^2(\gamma_j' x_i)+\sigma^2},$$

$$s_\beta^2 = \frac{\sigma^2 b_j^2\tau_j^2}{b_j^2\tau_j^2\sum_{i=1}^{n}\Psi^2(\gamma_j' x_i)+\sigma^2}.$$

$$[\sigma^2|\beta,\gamma,\mu,w,y]\sim IG\left(\frac{n+s}{2}, \frac{\sum_{i=1}^{n}(y_i-\beta_0-\sum_{j=1}^{m}\beta_j\Psi(\gamma_j' x_i))^2+S}{2}\right) \tag{3.12}$$

where $\beta=(\beta_1,\beta_2,\cdots,\beta_m)'$ and $\gamma_j=(\gamma_{j0},\gamma_{j1},\cdots,\gamma_{jp})'$.

$$\begin{aligned}[\gamma_{jh}\,|\,\gamma_{(jh)},\beta,\sigma^2,\mu,w,y] &\propto \exp\left[-\frac{1}{2}(\gamma_{jh}-\mu_h)^2\right] \\ &\times \exp\left[\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\beta_0-\sum_{j=1}^{m}\beta_j\Psi(\gamma_j' x_i))^2\right]\end{aligned} \tag{3.13}$$

where $\gamma_{(jh)}=(\gamma_{j0},\cdots,\gamma_{jh-1},\gamma_{jh+1},\cdots,\gamma_{jp})'$, $h=0,\cdots,p$.

$$\begin{aligned}[\mu_h|\gamma,\beta,\sigma^2,\ \mu_{(h)},w,y] = \\ N\left((m\sigma^{-2}\tau+A_\gamma^{-1})^{-1}(\sigma_\gamma^{-2}\sum_{j=1}^{m}\gamma_{jh}+A_\gamma^{-1}a_h),(m\sigma_\gamma^{-2}+A_\gamma^{-1})^{-1}\right)\end{aligned} \tag{3.14}$$

where $\mu_{(h)}=(\mu_0,\cdots,\mu_{h-1},\mu_{h+1},\cdots,\mu_p)'$, $h=0,\cdots,p$.

$$[w=w^{(k)}|\gamma,\beta,\sigma^2,\mu,y] = \frac{P_k^*}{\sum_{i=0}^{m}P_i^*}, \quad k=0,\cdots,m \tag{3.15}$$

where $P_0^* = \prod_{k=1}^{m}\frac{1}{\sqrt{\tau_k^2}}\exp\left[-\sum_{k=1}^{m}\frac{\beta_k^2}{2\tau_k^2}\right]\times\pi(w^{(0)})$,

$$P_j^* = \prod_{k=1}^{m}\frac{1}{\sqrt{b_k^2\tau_k^2}}\exp\left[-\sum_{k=1}^{m}\frac{\beta_j^2}{2b_k^2\tau_k^2}\right]\times\pi(w^{(j)}).$$

$w^{(0)}=(w_1=0,\cdots,w_m=0)'$, for $j=1,\cdots,m$,

$w^{(j)}=(w_1=1,\cdots,w_j=1,w_{j+1}=0,\cdots,w_m=0)'$, $b_k=\begin{cases}1 & \text{if } w_k=0\\ c_k & \text{if } w_k=1\end{cases}$. $\pi(w^{(0)})$ and $\pi(w^{(j)})$ are defined in (3.9).

In sampling scheme, the conditional distributions of (3.11), (3.12), and (3.14) are straightforward. But the conditional distribution of γ in (3.13) does not have intractable full conditional form, hence, we use the Metropolis algorithm (Metropolis et. al., 1953).

To implement the Metropolis algorithm, it is necessary that a suitable candidate generating density(Chib and Greenberg, 1995) be specified. For example, if $\pi(t)$ can be written as $\pi(t)\propto\psi(t)h(t)$, where $h(t)$ is a easily sampled density and $\psi(t)$ is uniformly bounded, then let $h(t)$ be the candidate generating function to draw candidates of γ. For this model, through (3.13), we set $h(\gamma_{jh})=N(\mu_h,1)$ and

$$\psi(\gamma_{jh})=\exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{m}\frac{\beta_j}{1+\exp(-\gamma_{j0}-\sum_{h=1}^{p}\gamma_{jh}x_{ih})}\right)^2\right]. \tag{3.16}$$

Given $\gamma_{jh}^{(l)}$ as the l-th iterate state, draw candidate γ_{jh}^{*} from $N(\mu_h,1)$, and accept γ_{jh}^{*} as $(l+1)$-th iterate state of γ_{jh} with acceptance probability $\alpha(\gamma_{jh}^{(l)},\gamma_{jh}^{*})=\min\left(\frac{\psi(\gamma_{jh}^{*})}{\psi(\gamma_{jh}^{(l)})},1\right)$ otherwise, reject the candidate and keep the current value of γ_{jh}. Finally, the joint conditional probability of (3.15) is obtained by Bayes theorem and this sampling is straightforward. The scheme goes through the sampling steps in (3.11)-(3.15) until the convergence is achieved.

4. Simulated Data

In this paper, we simulate one hidden layer which have 2 node as follows;

$$y_i=\beta_0+\beta_1\Psi(\gamma_1'x_i)+\beta_2\Psi(\gamma_2'x_i)+\varepsilon_i,\quad i=1,\cdots,n \tag{4.1}$$

where $\Psi(\gamma_1'x_i)=\frac{1}{1+\exp(-\gamma_{10}-\gamma_{11}x_{i1}-\gamma_{12}x_{i2})}$, and

$\Psi(\gamma_2'x_i)=\frac{1}{1+\exp(-\gamma_{20}-\gamma_{21}x_{i1}-\gamma_{22}x_{i2})}$.

Set $\beta_0=1.5$, $\beta_1=2.0$, $\beta_2=0.5$, $\gamma_{10}=0.1$, $\gamma_{11}=1.0$, $\gamma_{12}=2.0$, $\gamma_{20}=0.5$, $\gamma_{21}=3.0$ $\gamma_{22}=5,0$ and $\varepsilon_i\sim N(0,1)$, respectively.

Our simulation data of $n=200$ is obtained form (4.1). One choice of starting hyperparameters is as follows ;

$\sigma^{2\gamma} = I_p$, $a_\gamma = (0, \cdots, 0)'$, $A_\gamma = I_p$, $s = 0.0$, and $S = 0.0$. The reason for above setup are computational efficiency and that we have no information of prior. Thus we used noninformative prior, and set $c^2 = 100$, $\tau^2 = 0.01$, and $p_1 = p_2 = p_3 = p_4 = 0.5$. The Gibbs sampler generates 20000 iterations and Metropolis algorithm for generating γ is repeated 30,000 times, After discarding first 18,000 iterations, we use only the variates of remaining iterations. Convergence of the Gibbs sampler was assessed via Geweke (1992) method, using the CODA (Best, Cowles and Vines, 1995) suitable of diagnostics in S-plus. Fixed $m = 4$, the number of parameters is 20, such as $\beta_0, \cdots, \beta_4$, $\gamma_{10}, \cdots, \gamma_{42}$, $\mu_0, \cdots, \mu_4$ and σ^2. Most of the parameters had Geweke statistics between -1.96 and 1.96, we can decide that convergence is plausible.

Since $m = 4$, the possible values of w are $w^{(0)}, w^{(1)}, w^{(2)}, w^{(3)}$ and $w^{(4)}$ where $w^{(j)}$ is defined in (3.15) for $j = 0, \cdots, 4$ and is $w^{(j)}$ is generated from $[w = w^{(j)} | \gamma, \beta, \sigma^2, \mu, y]$ in (3.17). If $w^{(j)}$ is selected, this means that the model with j nodes is selected. Table 4.1 presents the marginal posterior probability of all possible models.

Table 4.1. The Number of Nodes

Number of Node	Frequency	Posterior prob. of w
0 $(w^{(0)})$	0	0
1 $(w^{(1)})$	515	0.2575
2 $(w^{(2)})$	1087	0.5435
3 $(w^{(3)})$	360	0.18
4 $(w^{(4)})$	38	0.019

From table 4.1, posterior probabilities of one node, two nodes, three nodes, and four nodes are 0.2575, 0.5435, 0.18 and 0.019, respectively. Therefore it is reasonable to choose the model with two nodes.

Table 4.2. Estimate β

Variable	Posterior mean	Posterior 95% interval estimate
β_1	1.7237267	(0.34390, 2.59721)
β_2	0.4099409	(-0.12952, 1.33900)
β_3	0.2535759	(-0.14442, 1.63406)
β_4	0.0199495	(-0.15307, 0.18844)

From Table 4.2, the posterior 95% interval of β_3 and β_4 contains "zero", but the posterior 95% interval of β_1 does not. This result corresponds to our expectation that the result of β_3 and β_4 would be "safely" estimated by 0, and β_1 would be generated around 2.0 (See figure 2).

From figure 2, we know that β_1 is generated around 2.0 and β_2 is generated around 0.5. Also β_3 and β_4 is estimated around 0. (The distribution of β_1, β_2, β_3, and β_4 are plotted as the real line, doted line, dashed line, and long-dashed line, respectively.)

5. Conclusion

In this thesis, Neural network models are studied in the view of Bayesian approach. Our concern is how many nodes should be in the model by latent variable. The idea of this model comes from the concept of Tanner and Wong(1987). Under the above circumstances, we have big concern about latent variables w which contains the information relevant to the number of nodes. The vector of latent variable, $w^{(j)}$, is interpreted as the statistician's prior probability which the $\Psi(\gamma_j' x_i)$'s correspond to non-zero components of w. Hence, we introduced the 'sequential step' for our model selection. As computational technique, the latent variable w was defined to simplify the joint posterior distribution. Gibbs sampler(Gelfand and Smith, 1990) and Metropolis algorithm(Metropolis et al., 1953) are employed to avoid the Bayesian difficult computation. Model which have two hidden nodes was simulated and applied to the proposed model. According to the experiment, this model was found to efficient to choose the number of nodes.

References

[1] Anderson, J. A. (1982). Logistic Discrimination. In *Classification, Pattern Recognition and Reduction of Dimensionality*, eds. P. R Krishnaiah and L.n. Kanal, Vol. of *Handbook of Statistics* 169-191. Amsterdam: North Holland.

[2] Best, N. G., Cowles, M. K. and Vines, S. K. (1995). *Convergence Diagnosis and Output Analysis Software for Gibbs Sampling Output, Version 0.3* Cambridge, MRC iostatistics Unit.

[3] Buntine, W. and Weigend, A. (1991). Bayesian back-propagation, *Complex Systems*, 5, 603-643.

[4] Chib, S. and Greenberg, E. (1995). Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, Vol 49, 327-335.

[5] Chung, Y. and Kim, H. (1999). Bayesian outlier detection in regression model. *Journal of Korean Statistical Society*, 28, 311-324.

[6] Gelfand, A.E.and Smith, A. F. M. (1990). Sampling-Based Approaches to Calculating Marginal Densities. *Journal of the American Statistical Association*, Vol 85, 398-409.

[7] George, E. I. and McCulloch, R. E. (1993). Variable Selection Via Gibbs Sampling. *Journal of the American Statistical Association*, Vol 88, 881-889.

[8] Geweke, J. (1992). Evaluating the Accuracy of Sampling-Based Approaches to Calculating Posterior Moments. *In Bayesian Statistics 4,* ed. J. M. Bernado. J. O. Berger, A. P. Dawid and A. F. M. Smith, Oxford, UK; Oxford University Press, 169-193.

[9] Mackay, D. J. C. (1992). Bayesian Methods for Adaptive Methods. Ph.D thesis, California Institute of Technology.

[10] McCulloch, W. S. and Pitts, W. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, 5, 115-133.

[11] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21, 1087-1092.

[12] Muller, P. and Rios Insua, D. (1998a). Feedforward Neural Networks for Nonparametric Regression. in *Practical Nonparametric and Semiparametric Bayesian Statistics*, eds. D. Dey, P. Muller, and D. Sinha. New York; Springer-Verlag.

[13] Muller, P. and Rios Insua, D. (1998b). Issues in Bayesian Analysis of Neural Network Models. *Neural Computation* 10, 571-592.

[14] Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. New York: Springer.

[15] Specht, D. F. (1991). A general Regression Neural Network. *IEEE Trans. Neural Networks*, 2, 568-576.

[16] Tanner, M. A. and Wong, W. H. (1987). The Calculation of Posterior Distributions by Data Augmentation. *Journal of the American Statistical Association*, Vol. 82, 528-550.

[2001년 6월 접수, 2002년 2월 채택]